Bangla Documents Analyzer

Samina Azad[1], Mohammad Shamsul Arefin[1], A.N.M. Rezaul Karim[2]
[1]Department of Computer Science and Engineering, Chittagong University of Engineering and Technology.
[2]Department of Computer Science and Engineering, International Islamic University Chittagong
e-mail: samina@yahoo.com, sarefin@cuet.ac.bd, zakianaser@yahoo.com

Abstract:

This paper presents the implementation of a system that analyzes Bangla documents to recognize its grammatical structures. The system takes a Bangla document as input. Then the system isolates all the individual sentences and extracts the words from each individual sentence eliminating the punctuation marks. A collection of stored knowledge and rules are applied to the words and sentences to detect grammatical features of the individual sentences. Finally the system performs different types of analysis from different aspects such as word detection, tense of verb recognition and types of sentence determination.

KEY WORDS: word code, word detection, tense detection, and type detection

1. INTRODUCTION

It was a great attempt of the computer scientists of the modern age to set up an intelligent machine that could understand human languages. But the earlier researches kept the attention only on English as the well-recognized international language. Recently some great efforts have been made to process some other languages. Bangla is a very rich language and approximately 10% of the people in the world use to speak in Bangla [1]. A most recent effort invoking Bangla is a system that takes Bangla statements written in English character set and displays it in Bangla characters. Unfortunately any of these systems is not able to provide sufficient aids for transformation of Bangla language to any others. Here in this purpose the need arises for understanding the linguistic distinctiveness of Bangla language. It is unconcerned so far to have any application software that grammatically analyzes Bangla documents or texts for notifying structural language attributes. In this paper we are representing a system to perform morphological analysis of Bangla statements and based on this analysis to signify some grammatical characteristics i.e. parts-of-speeches, tense of verbs, types of sentences etc. using artificial knowledge. This analysis is a precondition for linguistic transformation.

2. BANGLA DOCUMENT ANALYSIS

The system that analyses Bangla document- breaks it into easily perceivable units and finally makes some decisions based on simple heuristic knowledge. The system performs the tasks of isolating the sentences within it, detecting each of the parts of speeches in each sentence, determining the roots of verbs of those sentences, determining the types of verbs using the roots, notifing the specifiers and beevokties used in each sentence, findings out the numbers of the nouns used at that sentence, finding out the persons i.e. first person, second person and third person at that sentence, deciding the tense of verb for the sentence and finally categorizes the sentences in simple, complex and compound types through logical analysis.

For performing the task, the system performs morphological analysis of the sentences in the document. Hence, each word is categorized in its own type of parts-of-speeches and finite verb of the sentence is detected. This is done by operating on the root of the verb. The root is combined with some suffixes called – 'beevokties' to form the verb. The verb is then classified on the basis of these suffixes. The verb detects the tense of the sentence. If all the words in a sentence are not identified distinctively then the sentence is scanned from the right and if required from the left also. This scanning determines the left over undetected words. After this morphological analysis- the system syntactically checks the sentence to discover its type.

2.1 Representation of Bangla characters

In this system Bangla characters are represented using unicode conventions. The unicode display depends on the key typed on the keyboard. For example- if the user press the key "k" the character will be displayed as ' '. The keys on the keyboard also represent Kars and Folas. For example- if we type "a" it will be displayed as ' '. The compound letters in Bangla character set are represented by a combination of keys. Some examples are illustrated in Table-1.

Table 1: Examples of Compound Letters

Key Typed	Unicode Display
jnHm	জন্ম
Mit\Hr	মিত্র
AkHxr	অক্ষর

SkHt\i	শক্তি
JYHja	ঝাঞ্জা

2.2 Preprocessing of the text document

The sentences in the whole documents are isolated and each of the sentences is sent to the analyzer. A sentence always ends with punctuation like - '।','!','?'. The words of each sentence are then segregated for morphological analysis. Each word is terminated with a space or punctuation like - ' ,'':'',-'',.','।','?','!' .

2.3 Morphological Analysis

In morphological analysis of natural language processing- individual words are analyzed into their components and punctuations are separated from the words [5].

2.3.1 Word Detection

In this phase of the system individual words of the documents are classified according to their parts of speech invoking word detection method. First, the whole word is searched in the database. If the word is found then this method terminates immediately by setting detection = yes. If the word is not found then any substring of the word that is matches some words in the database is detected and the system checks if the prefix and the suffix of the word are valid. The prefix is accounted as valid if the prefix is found in the list of উপসর্গ. The suffix is valid if it is a modifier or any types of বিভক্তি as well as নির্দেশক. The নির্দেশক defines the number of the nouns. বিভক্তি denotes the case of the word in the sentence. There are two sorts of বিভক্তি - নাম বিভক্তি and ক্রিয়া বিভক্তি. নাম বিভক্তি is placed after the nouns and ক্রিয়া বিভক্তি is placed after the verbs. Some examples [3] are listed in Table-2.

Table-2: Formats of 'Beevokty' and 'Nirdesok'

বচন	নাম বিভক্তি	ক্রিয়া বিভক্তি	নির্দেশক
একবচন	রহিম+এর = রহিমের	খেল+এ = খেলে	লোক+টা = লোকটা
	নদী+তে = নদীতে	যা+ছিল = যাচ্ছিল	রুটি+খানা = রুটিখানা
বহুবচন	রবীন্দ্র+দেরকে = রবীন্দ্রদেরকে	চল+ছে = চলছে	পর্বত+মালা = পর্বতমালা
	মানুষ+দের = মানুষদের	বল+এছিলেন = বলেছিলেন	তারকা+রাজি = তারকারাজি

The different formats of the prefix and suffix combination are listed in Table-3

Table-3: Different formats of suffix and prefix

উপসর্গ	বিভক্তি	নির্দেশক	উপসর্গ + বিভক্তি	উপসর্গ + নির্দেশক	উপসর্গ + বিভক্তি + নির্দেশক
উপর	মাঝের	লোকটা	উপহারের	প্রভাতখানি	উপহারটার
অজানা	মধুর	মুখখানি	অজানার	অপসারণলো	প্রভাতখানার
সুনীল	হাসির	চাবাটে	সুনীলের	অসুখটা	সুনীলের

Sometimes the word contains a modifier at the end as a suffix to modify the original word as in - মমতাময়, ধৈর্যশীল, মানবতা, গতিশীলতা etc. The flowchart of the word detection module is given in Figure-1.

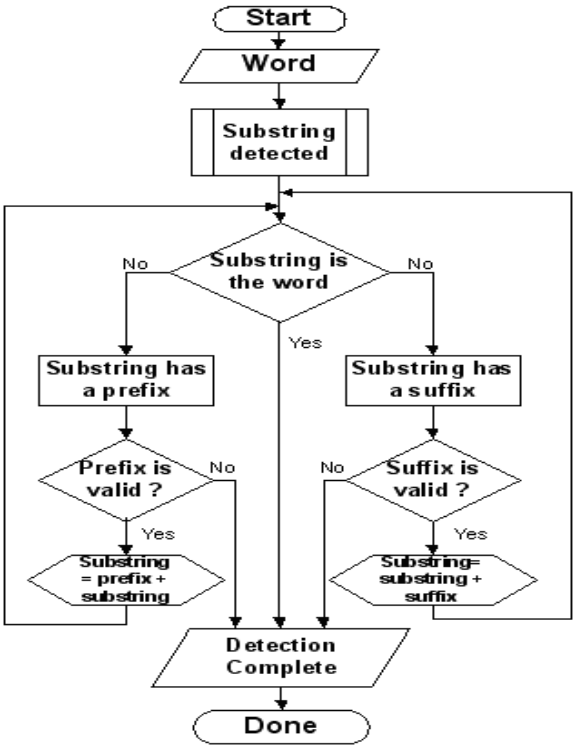

Figure-1: The word detection module

2.3.2 Verb Detection

Roots of the verbs are detected as in the same way described in the above section. The only difference is – there are some extra processing tasks. The roots of Bangla verbs are categorized into twenty classes. While mapping the words the root is detected to signify its class. Each class has its own list of ক্রিয়া বিভক্তি. The classes of roots are listed in Table-4 [2].

Table-4: Classes of the Roots

গণ (Class)	ধাতু (Root)
হ্	হ্ (হওয়া) , ল (লওয়া)
খা	খা (খাওয়া) , পা (পাওয়া) , যা (যাওয়া)
দি	দি (দেওয়া) , নি (নেওয়া)
শু	চু (চোঁয়ান) , নু (নোয়ান) , ছু (ছোঁয়া)
কর্	কর্ (করা) , কম্ (কমা) , চল্ (চলা)
কহ্	কহ্ (কহা) , সহ্ (সহা) , বহ্ (বহা)
কাট্	গার্থ , চাল , আক্ , বাঁধ , কাঁদ
গাহ্	চাহ্ , বাহ্ , নাহ্
লিখ্	কিন্ , ঘির্ , জিত্ , ফির্ , ভিড়্ , চিন্
উঠ্	উড়্ , শুন্ , ফুট্ , খুঁজ্ , খুন্ , ডুব্ , তুল্

After detecting the class of the root, the system checks the list of beevokties only for that class making it a heuristic technique. If any of the beevokties matches with the suffix of that word then the system sets word detection = yes.

2.3.2 Tense of Verb Detection

Our system detects the tense of verb while going through the verb detection process. Each beevokty in the list has a tense code. When the system gets a match between the suffix of the original word and an element in the list of ক্রিয়া বিভক্তি, it immediately collects the tense code attached to that element.
The flowchart of tense detection module is given in Figure-2.

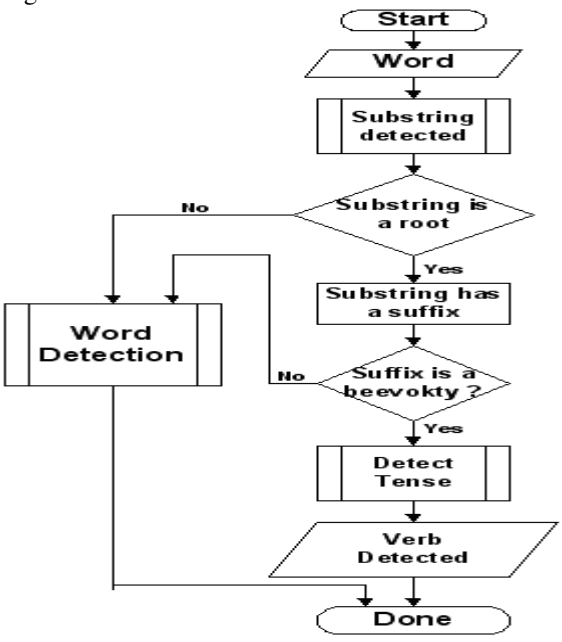

Figure-2: Tense of verb detection module

For example- if the original word is করছিলেন. The system first checks this word in the database and detects the root - কর. This root is of class- কর and so the system will check the list of ক্রিয়া বিভক্তি only for this class. Finally the ক্রিয়া বিভক্তি will be found to be ছিলেন. From the predefined stored knowledge the system finally give a decision that the sentence is in Past Continuous tense.

2.4 Syntactic Analysis

By syntactic analysis- linear sequences of words are transformed into structure that shows how words are related to each other. Some word sequences may be rejected if they violate the language rules for how words may be combined [5].

2.4.1 Exclusion of Ambiguity

In this system we have represented each type of parts-of-speech by assigning a word code to each word. This code is unique for each type of parts of speech. So, by checking the word code we can easily identify the parts-of-speech. The word code assignment scheme is depicted in the Figure-3.

Figure-3: Word Code Assignment Scheme

Very often the system found some ambiguity while determining the parts-of-speech. Because the same word in the database may be stored as different type of parts-of-speech and in this case the system need to invoke some extra efforts. This effort tends the system to have a more detection module, which tries to obtain the correct word code by observing the co-relations of the words in the sentence.

For example: If in any sentence the system gets the word 'সত্য' then it immediately checks the word in the database and detect it as a বিশেষ্য as well as বিশেষণ. Hence, the word code will be [1+2] = 3 (From Figure-3) which is an ambiguous word code.

সত্য কথা বল।	সত্য - বিশেষণ
এ এক বিরাট সত্য।	সত্য - বিশেষ্য

To avoid this situation our system applies some rules as stated in the following.
Let, n is the total number of words in the sentence and w[i] be the words in the sentence- where i = 1, 2, 3, 4, ……., n. Then the word code will be corrected according to the following assumptions shown in Table-5 and Table-6 considering different cases.

// Forward Scanning from right to left
Table-5: Test conditions for words

Case 1: wordCode[i]=3		
Priorities of conditions	Conditions to check	Updates
1	i>1 & (i=n-1) & wordCode[i-1]=2	wordCode[i]=1
2	wordCode[i+1]=16	wordCode[i]=1;
3	wordCode[i+1]=32	wordCode[i]=1
4	i>0 & (wordCode[i-1]=1 | wordCode[i-1]=4) & (wordCode[i+1]=1 | wordCode[i+1]=4)	wordCode[i]=2
5	i=0 & wordCode[i+1]=1	wordCode[i]=2
6	i=0	wordCode[i]=1
Case 2: wordCode[i]=10		
Priorities of conditions	Conditions to check	Updates
1	wordCode[i+1]=1 | wordCode[i+1]=2	wordCode[i]=2
2	Null	wordCode[i]=8
Case 3: wordCode[i]=17 | wordCode[i]=18 | wordCode[i]=19		
Priorities of conditions	Conditions to check	Updates
1	i>0 & wordCode[i-1]=4	wordCode[i]=16
2	i>0 & wordCode[i-1]=1	wordCode[i]=1
3	wordCode[i+1]=2	wordCode[i]=1
4	Null	wordCode[m]=1
Case 4: wordCode[i]=33 | wordCode[i]=35		
Priorities of conditions	Conditions to check	Updates
1	wordCode[i-1]=2	wordCode[i]=1
2	wordCode[i-1]=1	wordCode[i]=32
3	wordCode[i-1]=3	wordCode[i]=32 & wordCode[i-1]=1
4	wordCode[i-1]=4	wordCode[i]=33
5	wordCode[i-1]=8	wordCode[i]=32

// Reverse Scanning from left to right
Table-6: Test conditions for words

Case : wordCode[i]=3		
Priorities of conditions	Conditions to check	Updates
1	i>2 & wordCode[i-1]=1& wordCode[i-2]=1	wordCode[i]=2
2	i>2 & wordCode[i-1]=1 & wordCode[i-2]=4	wordCode[i]=2
3	i>2 & wordCode[i-1]=4 & wordCode[i-2]=1	wordCode[i]=2
4	i>1 & wordCode[i-1]=1 & wordCode[i+1]=1	wordCode[i]=2
5	i>1 & wordCode[i-1]=4 & wordCode[i+1]=1	wordCode[i]=2
6	i>1 & wordCode[i-1]=1 & wordCode[i+1]=4	wordCode[i]=2
7	i=n-1	wordCode[i]=1
8	i=1 & wordCode[i+1]=1	wordCode[i]=2

2.4.2 Types Of Sentence Detection

After all the words have been detected unambiguously the system scans the whole sentence and observes the co-ordination of the words to determine its type. There are three types of sentences [4] simple sentence (সরল বাক্য), complex sentence (জটিল বাক্য) and compound sentence (যৌগিক বাক্য) The test conditions for the three types of sentences to be checked are listed below-

Simple sentence: The sentence will contain only one finite verb or সমাপিকা ক্রিয়া and/or other non-finite verbs or অসমাপিকা ক্রিয়া that is only one word will have the word code = 32. If there is any conjunction in the sentence like এবং , আর , ও , বা , অথবা ; then the conjunction will connect two words of similar types i.e. the word code for word[i-1] will be same as that for word[i+1] where the conjunction is the ith word of the sentence. The conjunction in the sentence must not connect two verbs i.e. two words of word code 16 or 32. If the sentence contains any or some comma (,), then the commas will connect the words of the same word code. The comma will not connect two words of word code 16 or 32.

Complex sentence: The sentence contains more than one finite verb and/or other non-finite verbs i.e. at least two words with word code = 32 and may or may not have some other words with word code = 16. There will be no conjunction in the sentence such that it connects two words as stated in the simple sentence section rather then two clauses. The sentence may contain comma (,). The comma can conjunct two clauses, but the first clause will have a transitive verb with no object in that clause i.e. no word in the first clause will have word code = 1. If the sentence contains pair are words like - যা-তা,যে-সে,যিনি-তিনি,যেন-তেন,যখন-তখন,যেখানে-সেখানে etc. then the sentence will be complex. If the sentence encloses words like - যদি,যদিও,তবু,তবুও,কিনা etc. then the sentence will be complex sentence.

Compound sentence: The sentence contains more than one finite verb and/or other non-finite verbs i.e. at least two words with word code = 32 and may or may not have some other words with word code = 16.The sentences can include some specific words such as - তাই,অতএব,কিন্তু,তথাপি,সুতরাং,বরং,বরঞ্চ,পরন্তু,নইলে etc .The sentence can have some conjunctions- এবং , আর , ও , বা , অথবা etc. but connecting two clauses rather then two words i.e. the word code for word[i-1] will be same as that for word[i+1] where the conjunction is the ith word of the sentence. The conjunction in the sentence may connect two verbs i.e. two words of word code 16 or 32. If the sentence contains any or some comma (,), then the commas can make a contact between two or more clauses rather than words. If any clause includes a transitive verb then it must also include an object of word code = 1.

3 PERFORMANCE ANALYSIS

To make a measurement of the accuracy of the system we have incorporated some arbitrary sample documents and different kinds of analyzing process such as- word detection, tense of verb determination and types of sentence detection are applied on them. We also detected all these outcomes manually to verify the accuracy of the system. The performance of the system is then considered by comparing these two results and it is expressed in percentage.

3.1 The Word Detection Process

Words are searched in the database and if any ambiguity found then rules in Table-5 are applied to avoid them. Some arbitrary samples are analyzed by the system and the results are summarized in Chart-1.

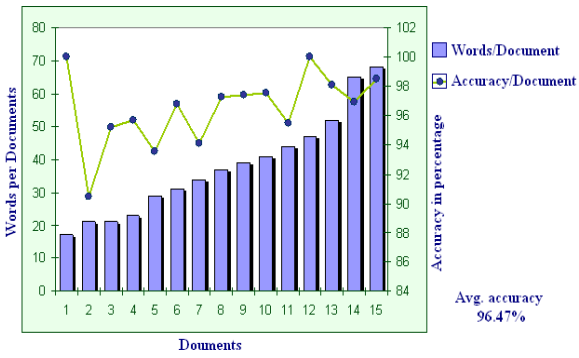

Chart-1: Experimental Results of Word Detection

The percentage accuracy will be improved gradually with the amount of input. This accuracy can be improved by adding some additional rules in the syntactic analysis of the system.

3.2 The Tense Detection Process

The tense of the sentence is detected on the basis of the main verb of the sentence. The tense detection module presents an *accuracy of 98%* with 15 samples. A portion of the experiment is depicted in Chart-2.

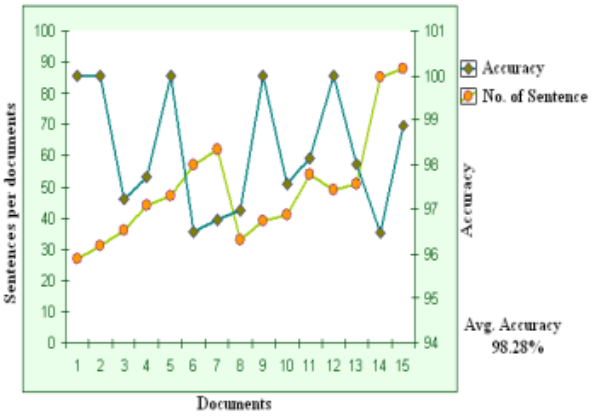

Chart-2: Experimental Results of Tense Detection

3.3 The Type Detection Process

The types of sentences are detected in the syntactic analysis of the system. The results with 15 samples are depicted in chart-3.

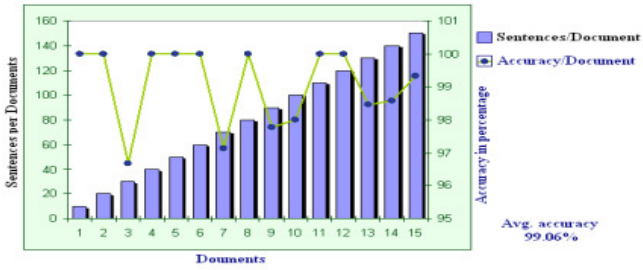

Chart-3: Evaluation of Type Detection

The experiment shows an average accuracy of 99%, however- the performance can be improved by adding syntactic rules.

4 CONCLUSIONS

In our system Bangla statements are represented using unicode character representation, so the system is machine independent and need not to utilize any specialized keyboard like bijoy. Often it is very troublesome to type the whole documents and the user can load an existing text document from the system and operate on it. The system we have represented is based on some knowledge in the database and some rules defined by the system. The system will fail to detect a word if it does not exist in the database. In this regard- we have enclosed some procedures to allow new entries in the vocabulary. The authority can *add*, *delete* and *modify* the databases. The authority can also provide some references to various words. The performance of the system can be improved by updating the detection modules of the system.

REFERENCES

[1] A. Tanvir, M. F. Zibran, R. Shammi and M. A. Sattar, "*Computer Representation Of Bangla Characters And Sorting of Bangla Words*". Proceedings of International Conference on Computer and Information Technology, 27-28 December 2002, Dhaka, Bangladesh.

[2] S. M. Chowdhury, S. M. H. Chowdhury, I. Khalil, K. D. Muhammod and S. Lahiri. "*Bangla Vashar Beyakoron*"

[3] M. Hoque "*Bangla Vashar Beyakoron and Rochonaritee*".

[4] M. Asadujjaman and S.H. Rahman. "*Ucchotoro Bangla Beyakoron and Rochona*".

[5] Rich and Night "*Artificial Intelligence*".

[6] Deitel and Deitel "How To Program JAVA".